

Distribution Arithmetic for Stochastic Analysis

Markus Olbrich, Erich Barke
Institute of Microelectronic Systems
Leibniz University of Hannover
Email: {markus.olbrich, eb}@ims.uni-hannover.de

Abstract—This paper presents a novel arithmetic which allows calculations with fluctuating values. The arithmetic consists of a special representation of random variables and procedures for performing numerical operations between them. Given the distributions of initial random variables, the moments (such as expected value, variance and higher moments) of any calculated variable can be determined. Our approach is not limited to normal distributions and works with linear and nonlinear functions. Correlations between variables are taken into account automatically by the arithmetic. Examples show the accuracy and runtimes compared to Monte Carlo simulation.

I. INTRODUCTION

In modern semiconductor process technologies relative parameter variations grow. In order to avoid both overdesign and low yield many approaches for circuit analysis have been investigated taking variations into account.

The general problem is computing the distribution of a performance value f (e.g. delay or power consumption) which is a function of a vector $\vec{x}$ of parameter values x_i (e.g. wire width or V_{th}) the distributions of which are known.

Range arithmetics like Affine Arithmetic have been successfully applied to EDA problems with varying parameters (e.g. [1]–[3]). They guarantee the inclusion of fluctuating values within calculated intervals or ranges. However, they suffer from the fact that the distributions inside the intervals are unknown. Therefore, yield estimation is not possible. Only by assuming linear operations and normal distributions can the interval sizes be interpreted as multiples of the standard deviation σ [3].

Monte Carlo simulation is the most general method for analyzing the distribution of performance values. It has (nearly) no limitations but it is very time consuming. Monte Carlo simulation serves as a reference when the accuracy of other methods is evaluated. Variance reducing methods, such as importance sampling, reduce the number of samples which speeds up Monte Carlo simulation significantly [4]. However, appropriate sampling densities have to be determined, which limits the general applicability.

Analytical approaches are faster by orders of magnitude. One way of doing so is to linearize the performance functions. This allows us to assume normal distributions

as results. However, in many cases a linear approximation leads to unacceptable errors.

Response Surface Methods (RSM) approximate performance functions by polynomials typically up to the second order [5]. This method leads to very fast moment calculation in comparison to Monte Carlo simulation.

Linearization and RSM are black box approaches. This means that the performance function is analyzed and approximated in a preprocessing step. In contrast, we focus on an analytical white box approach where the performance function is given by a formula or a program. The advantage of our method is that no preprocessing is necessary. Distributions of random variables are calculated easily by exchanging the normal real number arithmetic with the Distribution Arithmetic. Correlations between variables are treated automatically. Especially in the context of executable specifications (e.g. SystemC-AMS) our approach enables the calculation of value distributions very easily.

II. DISTRIBUTION ARITHMETIC

In this section, we define our Distribution Arithmetic (DA). It consists of a representation form for random variables and operations on these variables. DA variables are represented by a sum of the nominal value and weighted powers of initial random variables (IRVs) Δ_i . Additionally, mixed products of IRVs are allowed. Without loss of generality, the expected value of each IRV Δ_i is assumed to equal zero and each pair of IRVs is stochastically independent. This is no limitation because correlated variables can be represented by sharing the same IRV.

Following the preceding explanations the general representation form of a variable X is

$$X = \sum_{\vec{i} \in D} \left(x_{i_1, \dots, i_n} \prod_{j=1}^n \Delta_j^{i_j} \right) \quad (1)$$

with

$$D = \left\{ \vec{i} \in \mathbb{N}^n \mid \sum_{j=1}^n i_j \leq l \right\}. \quad (2)$$

We call the right hand side of Eqn. (1) a Distribution Arithmetic Form (DAF). The number of IRVs is n . The

order of the distribution arithmetic is l . The nominal value of X is $x_{0,\dots,0}$.

Example: If the number of IRVs is $n = 2$ and the order is $l = 3$ then a variable X is

$$\begin{aligned} X = & x_{3,0}\Delta_1^3 \\ & + x_{2,0}\Delta_1^2 + x_{2,1}\Delta_1^2\Delta_2 \\ & + x_{1,0}\Delta_1 + x_{1,1}\Delta_1\Delta_2 + x_{1,2}\Delta_1\Delta_2^2 \\ & + x_{0,0} + x_{0,1}\Delta_2 + x_{0,2}\Delta_2^2 + x_{0,3}\Delta_2^3, \end{aligned} \quad (3)$$

where $x_{0,0}$ is the nominal value.

Example: The wire width W of an interconnect wire in an integrated circuit is a sum of the nominal value w_0 and a deviation part $\sigma_W\Delta_1$ with constant σ_W and a distributed IRV Δ_1 . Assuming that Δ_1 is standard normal distributed, σ_W is the standard deviation of W . W can be expressed as a DAF in the following way:

$$W = w_0 + \sigma_W\Delta_1 \quad (4)$$

A. Unary Operations

It is a requirement that the result of any operation on a DA variable is again represented as a DA variable in the form of Eqn. (1). To meet this requirement, we approximate unary functions $T(X)$ by a Taylor Series up to the l -th order:

$$T(X) \approx \sum_{i=0}^l \frac{T_{Xi}(x_0)}{i!} (X - x_0)^i \quad (5)$$

with

$$T_{Xi}(x_0) := \left. \frac{\partial^i}{\partial X^i} T(X) \right|_{x_0}. \quad (6)$$

If a power of IRVs greater than l occurs during Taylor Series evaluation the corresponding coefficient is neglected. This assures that the order of the DA remains constant. Every arithmetical operation keeps the number of coefficients in Eqn. (1) constant.

Example: Let

$$X = x_0 + a\Delta_1 + b\Delta_1^2 + c\Delta_2 + d\Delta_1\Delta_2. \quad (7)$$

The Taylor Series (5) in the case of a second order DA is

$$\begin{aligned} T(X) \approx & T(x_0) \\ & + T'(x_0)(X - x_0) \\ & + \frac{T''(x_0)}{2!}(X - x_0)^2. \end{aligned} \quad (8)$$

For calculating the logarithm of X the first two derivatives are to be determined at the nominal point x_0 .

$$T(x_0) = \ln(x_0) \quad (9)$$

$$T'(x_0) = \frac{1}{x_0} \quad (10)$$

$$T''(x_0) = -\frac{1}{x_0^2} \quad (11)$$

Inserting into Eqn. (8) leads to

$$\begin{aligned} T \approx & \ln(x_0) + \frac{1}{x_0}(a\Delta_1 + b\Delta_1^2 + c\Delta_2 + d\Delta_1\Delta_2) \\ & - \frac{1}{2x_0^2}(a^2\Delta_1^2 + \cancel{2ab\Delta_1^3} + 2ac\Delta_1\Delta_2 \\ & + \cancel{2ad\Delta_1^2\Delta_2} + \cancel{b^2\Delta_1^4} + \cancel{2bc\Delta_1^2\Delta_2} \\ & + \cancel{2bd\Delta_1^3} + 2c^2\Delta_1^2\Delta_2^2 + c^2\Delta_2^2 \\ & + \cancel{cd\Delta_1\Delta_2^2} + \cancel{d^2\Delta_1^2\Delta_2^2}) \end{aligned} \quad (12)$$

where summands with added IRV powers higher than two are canceled. Equation (12) can be represented as a DAF (see Eqn. (1)):

$$\begin{aligned} T = & t_{2,0}\Delta_1^2 \\ & + t_{1,0}\Delta_1 + t_{1,1}\Delta_1\Delta_2 \\ & + t_{0,0} + t_{0,1}\Delta_2 + t_{0,2}\Delta_2^2 \end{aligned} \quad (13)$$

with

$$t_{0,0} = \ln(x_0) \quad (14)$$

$$t_{1,0} = \frac{a}{x_0} \quad (15)$$

$$t_{2,0} = \frac{b}{x_0} - \frac{a^2}{2x_0^2} \quad (16)$$

$$t_{0,1} = \frac{c}{x_0} \quad (17)$$

$$t_{0,2} = \frac{c^2}{2x_0^2} \quad (18)$$

$$t_{1,1} = \frac{d}{x_0} - \frac{ac}{x_0^2} \quad (19)$$

B. Binary Operations

One approach to treat binary operations like $T(X, Y)$ is to expand a Taylor Series in two dimensions similarly as in the previous section. However, this leads to a quadratic number of partial derivatives. Therefore, we reduce each binary operation to a multiplication or addition and unary operations. Examples are:

$$\frac{x}{y} = x(y^{-1}) \quad (20)$$

$$x^y = e^{y \log x} \quad (21)$$

This results in a much faster evaluation.

C. Moment Calculation

As mentioned above, we do not calculate probability density functions (PDFs) explicitly, but we determine the moments of distributions. If a PDF is required explicitly, then solutions for the so-called moment problem can be applied [6]. However, this is not the topic of this paper.

Moments are integral properties of distributions. Different kinds of moments exist. A distinction is drawn

between *raw moments* and *central moments*. The k -th raw moment of a random variable X is

$$m_{X,k} = E(X^k). \quad (22)$$

The first raw moment $m_{X,1}$ is the *expectancy value* $\mu_X = E(X)$ which is different from the nominal value in general. The k -th central moment is defined as

$$\mu_{X,k} = E((X - \mu_X)^k). \quad (23)$$

The second central moment is the *variance* σ_X^2 . Any central moment can be determined from raw moments [7]. Further distribution properties are *skewness* $\gamma_1 = \mu_3/\mu_2^{3/2}$ and *kurtosis* $\delta_2 = \mu_4/\mu_2^2 - 3$ which characterize the shape of the PDF (Probability Density Function).

Raw moments can easily be determined from the moments of the IRVs using the following relations.

$$E(\Delta_i^k) = m_{\Delta_i,k} \quad (24)$$

$$E(\Delta_i^k \Delta_j^l) = m_{\Delta_i,k} m_{\Delta_j,l} \quad (25)$$

$$m_{\Delta_i,0} = 1 \quad (26)$$

$$m_{\Delta_i,1} = 0 \quad (27)$$

The moments $m_{\Delta_i,k}$ of the IRVs are inputs to the Distribution Arithmetic. These moments allow to consider arbitrary IRV distributions. As a default, we assume standard normal distributions for IRVs. However, these moments can be determined directly from the statistical analysis of process variables without estimating the PDF.

D. Sensitivities

Since DAFs preserve the IRVs, the relation between deviations of resulting performance values and the initial random values can be determined without any additional effort. The dependencies can be analyzed by examining sensitivities which are the partial derivatives of a calculated value X with respect to the IRVs Δ_i . A sensitivity S_{Δ_i} equals the coefficient before the according Δ_i :

$$S_{\Delta_i} = \left. \frac{\partial X}{\partial \Delta_i} \right|_{\Delta_i=0} = \underbrace{x_{0,\dots,0,1,0,\dots,0}}_i \quad (28)$$

The sensitivities indicate which IRV has the strongest impact on the distribution of the performance values.

E. Semi-Symbolic Calculation

Whenever numerical operations are performed on DAF variables, IRVs are never replaced by actual values but the coefficients are evaluated numerically. This assures that the result of any operation is a DAF. Instead of applying a symbolic math system, we use a much faster special implementation which is called semi-symbolic. The symbols Δ_i are not represented explicitly but are considered implicitly in the implementation of numerical operations. The details are explained in the next section.

F. Implementation

The DA was implemented in C++ using operator overloading. Since the IRVs are global constants, only the coefficients $x_{i_1, \dots, i_n}$ are stored for each variable. These coefficients build an n -dimensional array. The IRV powers are not explicitly stored. They arise from the coefficient's position inside the array. In contrast to Affine Arithmetic the memory requirement for a variable does not depend on the number of operations that are performed. It is constant for a constant number of IRVs and DA order.

All standard mathematical functions were implemented. The central and raw moments, skewness, and kurtosis of a variable can be determined directly. Additionally, methods for calculating the covariance and correlation coefficient of variable pairs exist.

III. RESULTS

This section shows some applications of the DA. In the following all IRVs Δ_i are standard normal distributed. However, as mentioned above, the DA is capable of treating arbitrarily distributed IRVs. Only the moments of the distributions have to be known.

A. Accuracy of Elementary Functions

The accuracy of the DA depends on the functions which are calculated. The sum operation and the multiplication with a non varying skalar value are performed without any error. The multiplication of values of the form $x_{nom} + \sigma_x \Delta_0$ causes no error when the DA order is greater than 1.

The elementary operation that causes the greatest error is the exponential function which is examined in the following. The expectancy value and the standard deviation of the function e^X are investigated. The function's argument is composed as

$$X = x_0(1 + 10\% \Delta_1) \quad (29)$$

with standard normal distributed Δ_1 . The standard deviation of X is

$$\sigma_X = 10\% |x_0| \quad (30)$$

which increases with the absolute value of x_0 . Fig. 1 and Fig. 2 show the relative error of the expectancy value and the standard deviation of e^X . The results show that the commonly used linear treatment (1st order) leads to a significant error in this case. By contrast, using higher orders of the DA results in a much smaller relative error.

The results of the square root function are examined in the following. The relative error stays nearly constantly independently of x_0 . Fig. 3 and Fig. 4 show the relative error compared to exhaustive Monte Carlo simulation.

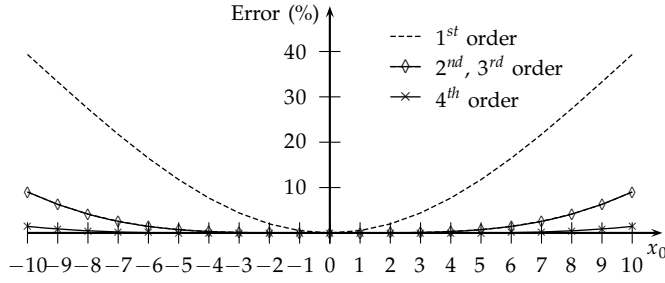


Fig. 1. Relative Expectancy Value Error of the Exponential Function

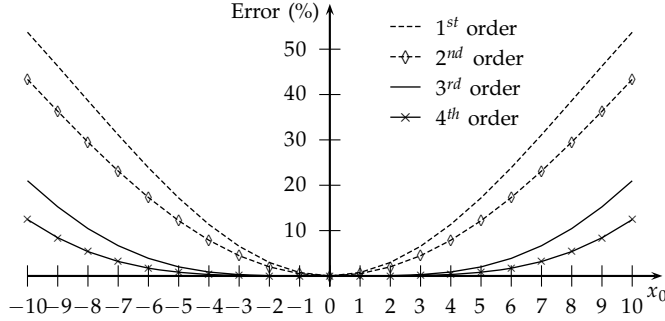


Fig. 2. Relative Standard Deviation Error of the Exponential Function

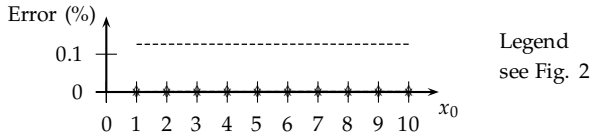


Fig. 3. Relative Expectancy Value Error of the Square Root

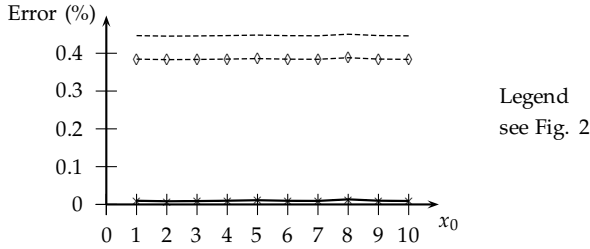


Fig. 4. Relative Standard Deviation Error of the Square Root

B. Elmore Delay of RC Tree

In order to demonstrate the use and the advantages of the Distribution Arithmetic the calculation of a wire delay affected by process variations is shown in the following section.

The DA is used to determine the distribution of an interconnect wire delay. A tree topology with one input and two output terminals is investigated. Fig. 5 shows the geometry and the wire segmentation which is used to build a lumped element model.

It is assumed that all wire segments lie in the same layer. The wire width w , its height h and its distance d

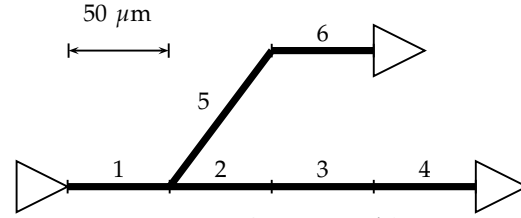


Fig. 5. Wire Geometry and Segmentation

from substrate are set to

$$w = (1 + 10\% \Delta_1) w_{\text{nom}} \quad (31)$$

$$h = (1 + 10\% \Delta_2) h_{\text{nom}} \quad (32)$$

$$d = (1 + 10\% \Delta_3) d_{\text{nom}} \quad (33)$$

As nominal values we set $w_{\text{nom}} = 0.14 \mu\text{m}$, $h_{\text{nom}} = 0.3 \mu\text{m}$, and $d_{\text{nom}} = 1 \mu\text{m}$. The values vary independently. The standard deviation σ is 10% of the nominal value for all geometry variables. The interconnect parameters are calculated from these values as follows.

The wire resistance R' per unit length is calculated by

$$R' = \frac{\rho}{wh}. \quad (34)$$

For calculating the wire capacitance C' per unit length we use the formula from [8]:

$$C' = \epsilon \left(\frac{w}{d} + \frac{\pi(1 - 0.0543h/2d)}{\ln \left(1 + \frac{2h}{d} + \sqrt{\frac{2h}{d} \left(\frac{2h}{d} + 2 \right)} \right)} + 1.47 \right) \quad (35)$$

For the input capacitances of the driven gates the following varying value is assumed:

$$C_g = (1 + 10\% \Delta_4) C_{g,\text{nom}} \quad (36)$$

The inner driver resistance R_D is assumed to be

$$R_D = (1 + 10\% \Delta_5) 100\Omega \quad (37)$$

This leads to the circuit model depicted in Fig. 6.

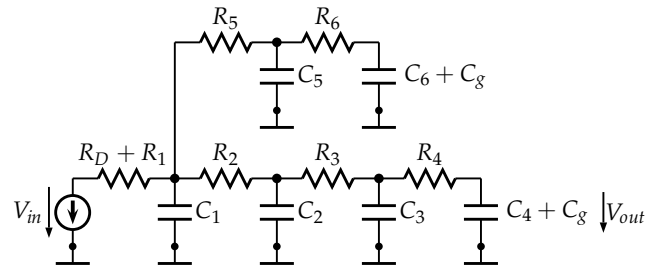


Fig. 6. RC Tree

In this example we apply the Elmore delay metric. However, any other delay metric could be used. The

TABLE I
RESULTS FOR THE RC-TREE DELAY τ ($\tau_{nom} = 4.1885 \cdot 10^{-11}$ s)

Order	μ_τ	σ_τ	Skewness	Kurtosis	$\text{cor}(R, C)$	Runtime
1	$4.1885 \cdot 10^{-11}$ s	$3.7230 \cdot 10^{-12}$ s	0.000	-1.036	0	0.04 ms
2	$4.2203 \cdot 10^{-11}$ s	$3.7401 \cdot 10^{-12}$ s	0.179	0.034	-0.5206	0.09 ms
3	$4.2203 \cdot 10^{-11}$ s	$3.7922 \cdot 10^{-12}$ s	0.191	0.106	-0.4909	0.34 ms
4	$4.2214 \cdot 10^{-11}$ s	$3.7944 \cdot 10^{-12}$ s	0.201	0.128	-0.5052	1.8 ms
MC 10^8	$4.2215 \cdot 10^{-11}$ s	$3.7977 \cdot 10^{-12}$ s	0.203	0.137	-0.5043	67 s

Elmore delay τ from V_{in} to V_{out} is calculated by

$$\begin{aligned} \tau = & (R_D + R_1)C_1 \\ & + (R_D + R_1 + R_2)C_2 \\ & + (R_D + R_1 + R_2 + R_3)C_3 \\ & + (R_D + R_1 + R_2 + R_3 + R_4)(C_4 + C_g) \\ & + (R_D + R_1)(C_5 + C_6 + C_g) \end{aligned} \quad (38)$$

The results are shown in Table I. Depending on the DA's order the results for the nominal delay τ_{nom} , the expectancy value μ_τ of the delay, the standard deviation σ_τ , the skewness γ_1 , the kurtosis γ_2 , and the correlation coefficient $\text{cor}(R, C)$ between R and C are shown. The correlation coefficient is defined as

$$\text{cor}(R, C) = \frac{E((R - \mu_R)(C - \mu_C))}{\sqrt{\sigma_R} \sqrt{\sigma_C}} \quad (39)$$

In the last row the results of a Monte Carlo simulation (MC) with 10^8 samples are given as a reference. The runtimes for evaluating the DAF of τ are taken from results on a 2.5 GHz Opteron PC under Linux.

Using the fourth order DA, the approach delivers results very close to the reference MC results. The runtime is below 2 ms which is orders of magnitudes away from the Monte Carlo simulation. Further speedup can be achieved at the cost of accuracy by reducing the DA order.

C. Gate Delay Metric

In this example, the gate delay for low voltage circuits is analyzed considering varying driver characteristics and supply voltage drop. In [9] the following gate delay metric is proposed based on the Alpha-power law [10]:

$$\tau \propto C_L \frac{3V_{DD}^3 + 3V_{DD}^2 V_T - 3V_{DD} V_T^2 + V_T^3}{(V_{DD} - V_T)^2 (V_{DD} - V_T)^\alpha} \quad (40)$$

In our investigation the threshold voltage V_T and the load capacitance C_L are varying values. Additionally, an IR-drop is considered by a stochastically independent variation of the supply voltage V_{DD} :

$$V_T = (1 + 20\% \Delta_1) V_{T,nom} \quad (41)$$

$$C_L = (1 + 5\% \Delta_2) C_{L,nom} \quad (42)$$

$$V_{DD} = (1 + 10\% \Delta_3) V_{DD,nom} \quad (43)$$

The following nominal values are used: $V_{T,nom} = 0.22$ V, $C_{L,nom} = 20$ fF, $V_{DD,nom} = 0.8$ V, and $\alpha = 1.5$.

Table II shows the results for different DA orders. Relative errors in comparison with a reference Monte Carlo simulation with 10^9 samples are shown in parenthesis. The results for the Monte Carlo simulation with 10^4 samples are also given. Since the resulting moments differ from run to run only the mean relative error of five MC simulation runs is shown for the MC 10^4 case. Although the runtime of the MC simulation is an order of magnitude higher than the 4th order DA, the error of the MC simulation is much higher.

In order to investigate which varying parameter has the highest impact on the resulting delay, we examine the sensitivities (see Section II-D).

$$S_{\Delta_1} = 2.0 \cdot 10^{-17} \quad (44)$$

$$S_{\Delta_2} = 3.6 \cdot 10^{-12} \quad (45)$$

$$S_{\Delta_3} = -3.6 \cdot 10^{-12} \quad (46)$$

S_{Δ_1} corresponds with V_T , S_{Δ_2} with C_L and S_{Δ_3} with V_{DD} , respectively. One can see that in this case the delay is much more sensitive against changes of the load capacitance and power supply variation than threshold voltage variation.

D. Runtimes of Single Operations

Since the number of value coefficients stays constant the runtime of any binary or unary operation is $O(1)$ for a constant number of IRVs.

Fig. 7 shows the runtimes of the binary operations + and * as well as the square root as an example of a unary operation. In general, the operations take advantage of zero entries, because nested loops with zero as a factor are skipped. The operations are applied to arguments with a fully defined coefficient matrix which is the worst case. Therefore, these runtime values are upper bounds. The runtimes of arithmetic operations strongly depend on the number of IRVs and the DA's order. The runtimes are essentially exponential with respect to the number of IRVs. However, for a limited number of IRVs the runtimes are very short and outperform Monte Carlo simulation. On the basis of Fig. 7 a tradeoff between runtime and accuracy can be found by choosing an appropriate order.

TABLE II
RESULTS FOR THE GATE DELAY METRIC (RELATIVE ERROR IN PARENTHESES)

Order	μ_τ	σ_τ	Skewness	Kurtosis	Runtime
1	$7.17138 \cdot 10^{-11}$ s (0.3 %)	$5.07094 \cdot 10^{-12}$ s (1.5 %)	0.0000 (100 %)	-1.500 (790 %)	0.00 ms
2	$7.19827 \cdot 10^{-11}$ s (0.0 %)	$5.08834 \cdot 10^{-12}$ s (1.2 %)	0.2618 (7.1 %)	0.001 (99.6 %)	0.01 ms
3	$7.19827 \cdot 10^{-11}$ s (0.0 %)	$5.14567 \cdot 10^{-12}$ s (0.1 %)	0.2709 (3.9 %)	0.177 (18.1 %)	0.04 ms
4	$7.19886 \cdot 10^{-11}$ s (0.0 %)	$5.14705 \cdot 10^{-12}$ s (0.1 %)	0.2806 (0.4 %)	0.207 (4.4 %)	0.13 ms
MC 10^4	(0.2 %)	(0.4 %)	(11.5 %)	(24.4 %)	6 ms
MC 10^9	$7.19887 \cdot 10^{-11}$ s	$5.14948 \cdot 10^{-12}$ s	0.2818	0.216	454 s

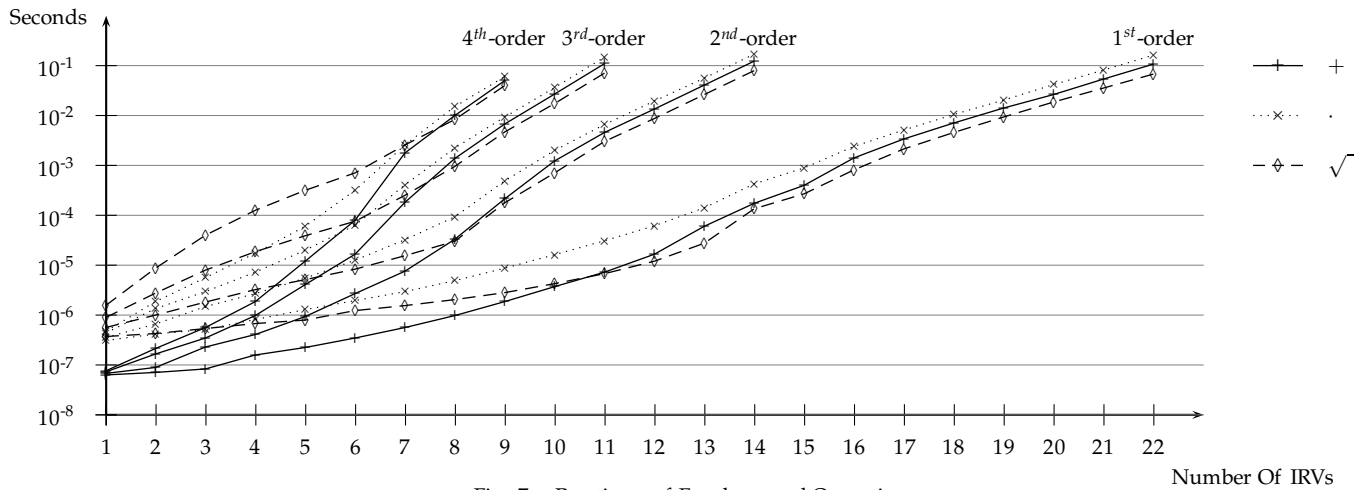


Fig. 7. Runtimes of Fundamental Operations

IV. CONCLUSION

We presented an arithmetic which allows to calculate with random values as simple as with real numbers. As a result, the distribution moments of any resulting value can be determined at any time. The memory requirement of the value representation is independent from the number of performed operations. The runtime of numerical operations is limited to constant values. The advantages of the approach are:

- Fast and accurate analysis of distribution moments for appropriate number of IRVs
- Easy use inside analysis environments due to encapsulation of stochastic details
- Correlations of even higher orders between random variables are considered
- Selectable tradeoff between runtime and accuracy by choosing DA order.

V. FUTURE WORK

In the future we want to insert the DA into an equation solver. This will allow us to perform DC and transient simulation directly using DA variables for the state and parameter variables.

REFERENCES

- [1] W. Heupke, C. Grimm, and K. Waldschmidt, "Modeling uncertainty in nonlinear analog systems with affine arithmetic," in *Advances in Specification and Design Languages for SoCs*, P. Boulet, Ed. Springer Verlag, December 2006.
- [2] J. Ma and R. Rutenbar, "Fast interval-valued statistical modeling of interconnect and effective capacitance," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 4, pp. 710–724, April 2006.
- [3] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar, "Probabilistic interval-valued computation: Toward a practical surrogate for statistics inside CAD tools," in *DAC 2006*, 2006, pp. 167–172.
- [4] M. Zhang, M. Olbrich, H. Kinzelbach, D. Seider, and E. Barke, "A fast and accurate Monte Carlo method for interconnect variation," in *ICICDT 2006*. IEEE, January 2006, pp. 207–210.
- [5] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 1, pp. 16–37, 2007.
- [6] J. A. Shohat and J. D. Tamarkin, *The Problem of Moments*. American mathematical society, New York, 1943.
- [7] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. McGraw-Hill Publishing Co., 2002.
- [8] T. Yuan, C.P.; Trick, "A simple formula for the estimation of the capacitance of two-dimensional interconnects in VLSI circuits," *Electron Device Letters*, 1982.
- [9] A. Ramalingam, S. V. Kodakara, A. Devgan, and D. Z. Pan, "Robust analytical gate delay modeling for low voltage circuits," in *ASP-DAC 2006*. New York, NY, USA: ACM Press, 2006, pp. 61–66.
- [10] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE Journal of Solid State Circuits*, vol. 25, no. 2, pp. 584–594, April 1990.